

## **REMARKS**

Claims 48-84 were presented for examination and were pending in this application. In an Office Action dated June 10, 2009, claims 48-84 were rejected. Claims 48, 55, 65 and 74 are amended herein. Reconsideration of the application in view of the above amendment and the following remarks is respectfully requested.

### **Response to Rejection Under 35 USC § 101**

Claims 48-84 are rejected under 35 USC § 101 as being directed towards non-statutory subject matter.

Claims 48-84 are rejected under 35 USC § 101 as being directed towards non-statutory subject matter. Specifically, the Office Action provides that: “the claimed subject matter provides for a final result of the method, system, computer program product of merely receiving a plurality of user inputs responsive to identifying the plurality of conflicts, each user input specifying ‘a conflict resolution procedure,’ particularly conflict resolution procedure does not have support from the specification [page 5-11].” (emphasis in original) *See* Office Action dated June 10, 2009, pg. 5. The Office Action also indicates that the claims do not “specifically define necessary hardware elements,” which prevents the claims from “achieving the required status of having ‘real world’ value.” *See* Office Action dated June 10, 2009; pgs. 5-6. This rejection is respectfully traversed in view of the amended claims.

Representative claim 48, as amended, recites “generating a third version of the associative array by merging modifications from the first version of the associative array and the second version of the associative array and resolving a plurality of conflicts between the first version of the associative array and the second version of the associative array by receiving a

plurality of user inputs responsive to identifying the plurality of conflicts, each user input specifying a conflict resolution procedure for an individual conflict, and storing the third version of the associative array in a memory.” (emphasis added) Support for this amendment is found throughout the specification, for example at paragraphs [0018]-[0020], [0046], [0049] and [0050]. Hence, amended claim 48 results in a memory which includes a third version of the associative array where conflicts between different versions of the associative array are resolved based on a user-specified conflict resolution procedure. Additionally, amended claim 48 is tied to an additional statutory class, an article (i.e., the memory which stores the third version of the associative array). This state of this article is further transformed by storing the third version of the associative array, which physically changes the memory. Accordingly, claim 48, as amended, results in a memory which stores a third version of the associative array where conflicts between a first version of the associative array and a second version of the associative array are resolved based on a user-specified conflict resolution procedure. Contrary to the assertion in the Office Action, independent claim 48 does not provide a “final result” of receiving user input, but results in a memory including a version of the associative array that has been modified based on the user-provided conflict resolution procedure. Accordingly, reconsideration and withdrawal of this rejection is respectfully requested.

Independent claims 55, 65 and 74 are similarly amended to more particularly describe that the third version of the associative array is stored in “a memory in communication with a processor,” clarifying that the “final result” of the independent claims is an article which has been transformed to include the third version of the associative array which has resolved conflicts between the first version of the associative array and the second version of the associative array based on a user-supplied conflict resolution process. As noted above with respect to claim 48,

support for these amendments is found throughout the specification, for example at paragraphs [0018]-[0020], [0046], [0049] and [0050]. Accordingly, amended claims 55, 65 and 74 now provide for a “final result” of a memory which stores a third version of the associative array where conflicts between a first version of the associative array and a second version of the associative array are resolved based on a user-specified conflict resolution procedure, so reconsideration and withdrawal of their rejection is respectfully requested.

As dependent claims 49-54, 56-61, 66-73 and 75-84 variously depend from independent claims 48, 55, 65 and 74, claims 49-54, 56-61, 66-73 and 75-84 also recite statutory subject matter for the reasons discussed above.

Further, it is respectfully noted that claims 65-69 recite components of an apparatus – “a data store” and a “version controller” – as well as the structural relationship between the components. Additionally, claims 74-89 recite a computer program product, and clearly specify that the computer program product comprises “a computer readable medium storing computer executable code for implementing version control on an associative array.” The necessary hardware component, the “computer readable medium,” applicable to claims 74-89 is specified in the claims, contrary to the assertion in the Office Action. *See* Office Action dated June 10, 2009, pgs. 5-6.

### **Response to Double Patenting Rejection**

Claims 48-84 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 24-28, 30-34 and 42-45 of co-pending Application No. 10/899,560. This rejection is respectfully traversed. It is indicated in the Office Action that a timely filed terminal disclaimer may be used to overcome the rejection. *See* Office Action dated August 11, 2008, pg. 7, ¶ 9. In the event that the rejection of claims 48-84 under

the judicially created doctrine of double patenting is maintained, it is requested that this rejection be held in abeyance until it is determined that there is allowable subject matter, at which time a terminal disclaimer will be filed.

Claims 48-84 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-14, 16 and 18-21 of co-pending Application No. 10/700,017. This rejection is respectfully traversed. It is indicated in the Office Action that a timely filed terminal disclaimer may be used to overcome the rejection. *See* Office Action dated August 11, 2008, pg. 7, ¶ 9. In the event that the rejection of claims 48-84 under the judicially created doctrine of double patenting is maintained, it is requested that this rejection be held in abeyance until it is determined that there is allowable subject matter, at which time a terminal disclaimer will be filed.

#### **Response to Rejection Under 35 USC § 103(a)**

In the 4th paragraph of the Office Action, Examiner rejected claims 48-61, 63-71, 73-80 and 82-84 as allegedly being unpatentable over U.S. Patent No. 6,631,386 to Arun et al. (“Arun”) in view of U.S. Patent No. 6,415,299 to Baisley et al. (“Baisley”). This rejection is overcome in view of the amended claims.

Claims 48, 55, 65 and 74 variously recite “generating a third version of the associative array by merging modifications from the first version of the associative array and the second version of the associative array and resolving a plurality of conflicts between the first version of the associative array and the second version of the associative array by receiving a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict, and storing the third version of the associative array in a memory.” (emphasis added)

Hence, the amended claims recite generating a third version of an associative array which incorporates changes from a first version of the associative array and a second version of the associative array. Generation of the third version of the associative array also resolves a plurality of conflicts between the first version of the associative array and the second version of the associative array by receiving a plurality of different user inputs. Each individual user input specifies a conflict resolution procedure applied to an individual conflict. Resolving a plurality of conflicts responsive to a plurality of different user inputs each individually identifying a conflict resolution procedure for a specific conflict beneficially allows greater user control over resolution of individual conflicts. For example, a first user input may specify that a first conflict be resolved by including data from the first version of the associative array in the third version of the associative array. Subsequently, a second user input may specify that a second conflict be resolved by including data from the second version of the associative array in the third version of the associative array. Hence, using a plurality of user inputs to resolve a plurality of conflicts beneficially allows independent resolution of individual conflicts between a set of two versions, allowing individual resolution of each conflict by selecting data from different versions. This allows increased customization of the third version of the associative array to include different data values from each version of the associative array.

Arun does not disclose at least the claimed element of “resolving a plurality of conflicts between the first version of the associative array and the second version of the associative array by receiving a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict.” Rather, Arun describes a database version control system where a database table includes “at least some” records having “a version control field including version control

information.” See Arun, col. 3, lines 58-61. Regarding the resolution of conflicts between versions, at most Arun describes a conflict resolution operation in which a user checking in a checked-out record version determines whether the checked-out record version or a record version stored in the database is preserved for later use. See Arun, col. 3, lines 17-20. Specifically, when a conflict exists, the user provides only a single user input which selects a single record version to resolve the conflict. See Arun, FIG. 6, item 153 and col. 17, lines 59-65. After this single user input, only the record version identified by the single user input survives after the conflict resolution operation, so all data from the selected version is maintained while all data from the non-selected version is discarded. See Arun, col. 3, lines 23-26.

Thus, for one or more conflicts between a first record version and a second record version, Arun merely provides a single user input that allows the user to choose between (a) resolving all conflicts between the two record versions with data from the first record version or (b) resolving all conflicts between the two record versions with data from the second record version. See Arun, FIG. 6 and FIG. 6A. Arun only allows for an additional user input if conflicts exist between one or more additional sets of record versions, in which case the single user input process repeats for each set until no conflicts exist between any set of two or more record versions. See Arun, FIG. 6A, item 158 and col. 8. While Arun resolves multiple conflicts, a single user input is used to resolve all of the identified conflicts by selecting a version which is used to supply the output for all identified conflicts.

Hence, Arun does not allow a user to individually resolve multiple conflicts by selecting data from different versions using user input for each conflict encountered, but rather uses a single user input to specify the version used to resolve all conflicts encountered. Therefore, Arun does not disclose at least the claimed element “resolving a plurality of conflicts between

the first version of the associative array and the second version of the associative array by receiving a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict.”

Baisley does not remedy the deficient disclosure of Arun. Rather, Baisley merges a source version of a model into a target version of the model in an object oriented repository. *See* Baisley, Abstract. A collection of versions occurring in a history of the source version is used to build a first list while a collection of versions occurring in a history of the target version is used to build a second list. The first and second lists are combined, either the first list or second list is identified as the non-preferred history and the first list, including the history of the source version, is added to the history of the target version. Baisley, col. 2, lines 10-23. Conflicts in the dual list of source version history and target version history are reported and resolved by adding the target version to a list of end versions on each conflicting attribute value in the non-preferred history. Baisley, col. 2, lines 25-27.

Baisley characterizes a conflict as occurring when the same object is modified in both lines of development, source and target. Baisley, col. 5, lines 37-41. However, rather than receive “a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict,” to resolve specific conflicts using a specific user input Baisley performs a predetermined conflict resolution procedure depending on the type of conflict. Baisley, col. 6, lines 16-23; Table II (at col. 6, line 25 to col. 7, line 15). For example if an object is deleted from one version and the object’s attribute is given a value in the other version, the object remains deleted. User input specifying a conflict resolution procedure for the conflict is not received, but a predefined action is taken to remedy the conflict. Baisley, col. 6, lines 27-32. As an additional example of conflict

resolution in Baisely, if a single-valued attribute is unset in one version and set to a new value in the other version, the new value is retained in the merged version. Baisley, col. 6, lines 36-42.

Hence, Baisley relies on predefined actions to resolve conflicts between different versions during a merge, and does not receive “a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict,” as claimed. The predefined actions in Baisley limit the ability of a user to control the source of the data used to resolve a conflict between versions by specifying execution of predefined actions to retrieve data from a version to resolve a conflict. At most, Baisely allows a user to select a preferred version of development, which does not allow for individual resolution of conflict, but sets a global parameter for the conflict resolution rules. After identifying a type of conflict, Baisley automatically applies a predefined conflict resolution rule associated with the conflict type to retrieve data from a version for inclusion in the merged version. Hence, rather than allow a user to individually select the version providing the data for the merged version, Baisley automates the data selection, removing user input from the conflict resolution process. There is no receipt of user input to resolve conflicts between versions in Baisely, much less “a plurality of different user inputs responsive to identifying the plurality of conflicts, each individual user input specifying a conflict resolution procedure for an individual conflict,” as claimed.

At most, Bailey and Arun disclose a system where a single user input is received and used to resolve all identified conflicts using data from a version identified by the single user input. While such a system is capable of resolving multiple conflicts, such a system does not allow user customization of how specific, individual conflicts are resolved. In contrast, the claimed invention receives a different user input for each conflict, allowing each conflict to be



resolved based on user input specific to individual conflicts. Resolving individual conflicts is not contemplated by Bailey or Arun, which at most permit a user to identify a version whose data is used to resolve all identified conflicts.

Thus, neither of the cited references, taken alone or in combination, teaches or suggests the claimed invention. Therefore, amended independent claims 48, 55, 65 and 74 are patentably distinct from the cited references, both alone and in combination, and withdrawal of the rejection is respectfully requested.

Claims 49-54 depend from claim 48, so all arguments advanced above with respect to claim 48 are hereby incorporated so as to apply to claims 49-54. Claims 56-64 depend from claim 55, so all arguments advanced above with respect to claim 55 are hereby incorporated so as to apply to claims 56-64. Claims 66-73 depend from claim 65, so all arguments advanced above with respect to claim 65 are hereby incorporated so as to apply to claims 66-73. Claims 75-84 depend from claim 74, so all arguments advanced above with respect to claim 74 are hereby incorporated so as to apply to claims 75-84. Therefore, claims 49-54, 56-64, 66-73 and 75-84 are patentably distinct from the cited references, both alone and in combination, and withdrawal of the rejection is respectfully requested.

**CONCLUSION**

In sum, Applicants respectfully submit that claims 48-84, as presented herein, are patentably distinguishable over the cited references (including references cited, but not applied). Therefore, Applicants request reconsideration of the basis for the rejections to these claims and requests allowance of them.

In addition, Applicants respectfully invite the Examiner to contact Applicants' representative at the number provided below if the Examiner believes it will help expedite furtherance of this application.

Respectfully submitted,  
Lawrence W. McVoy, et al.

Dated: December 10, 2009

By: /Brian G. Brannon/  
Brian G. Brannon, Registration No. 57,219  
Attorney for Applicants  
Fenwick & West LLP  
Silicon Valley Center  
801 California Street  
Mountain View, CA 94041  
Tel.: (650) 335-7610  
Fax: (650) 938-5200